VECTOR 04: TECHNICAL ARCHITECTURE — SYMBIOTIC CODES FRAMEWORK

Version: 1.1 → 1.2 **USS/CSI INTEGRATION**
Date: December 25, 2025
Vector: 4 of 23
Source: 18_270 Analysis + USS/CSI Framework v3.3

🆕 WHAT'S NEW IN v1.2

USS/CSI Integration:

1. **TDI_USS (Technical Debt Index):** Inverse metric for system health
2. **Security_USS:** Part of CSI (8% weight)
3. **HCI_USS Integration:** Human Capability Index linked to architecture
4. **USS Thresholds:** For all technical metrics
5. **Dashboard Integration:** Technical monitoring in USS format

New Sections:

- **Section 8:** Technical Metrics (USS)
- **Section 9:** Security_USS Specification
- **Section 10:** TDI_USS Specification
- **Section 11:** Technical Dashboard

Updated References:

- ANNEX Metrics Framework v3.3
- V12 Measurement Systems v1.1
- Integration Map v1.6

EXECUTIVE SUMMARY

Core Finding: Symbiotic architecture embeds partnership at training level. Multi-AGI coordination requires technical protocols AND unified measurement.

USS Integration: Technical health measured through:

- **Security_USS** (8% of CSI) — System protection

- **TDI_USS** — Technical debt monitoring

- **HCI_USS** — Human capability preservation

Critical Principle:

> Technical architecture must be measurable. If you can't measure it, you can't manage it.

SECTION 1-7: [PRESERVED FROM v1.1]

Sections 1-7 содержат AGI Communication Protocol, Trust Framework, Conflict Resolution, World Model Sync, Security Architecture, Entropy Detection, Human-Preservation Interface — см. полный текст в v1.1

SECTION 8: TECHNICAL METRICS (USS) □ NEW

8.1 Technical Metrics Overview

All technical metrics normalized to USS (0-1):

| Metric | Full Name | Formula | CSI Weight |

|--------|-----------|---------|------------|

| Security_USS | System Security Index | See 9.1 | 8% |

| TDI_USS | Technical Debt Index | 1 - TDI | — |

| Infra_USS | Infrastructure Health | See 8.2 | — |

| Cyber_USS | Cyber Defense Score | See 8.3 | — |

| Drift_USS | Entropy/Drift Index | 1 - Drift | — |

8.2 Infrastructure Health (Infra_USS)

```
Infra_USS = (Availability + Redundancy + Performance + Scalability) / 4

AVAILABILITY (0-1):
├─ Uptime ratio (actual/target)
├─ Target: 99.99% = 0.9999
├─ Example: 99.95% = 0.9995
└─ Threshold: <0.999 = WARNING

REDUNDANCY (0-1):
├─ Geographic distribution score
├─ Backup coverage ratio
├─ Failover capability
└─ Target: 0.95+

PERFORMANCE (0-1):
├─ Response time ratio (target/actual)
├─ Throughput ratio
├─ Latency inverse
└─ Target: 0.90+

SCALABILITY (0-1):
├─ Capacity headroom
├─ Elastic scaling speed
├─ Peak handling capability
└─ Target: 0.85+
```

8.3 Cyber Defense Score (Cyber_USS)

```
Cyber_USS = (Prevention + Detection + Response + Recovery) / 4

PREVENTION (0-1):
├─ Vulnerability patching rate
├─ Attack surface minimization
├─ Access control effectiveness
└─ Encryption coverage

DETECTION (0-1):
├─ Threat detection rate
├─ False positive rate (inverse)
├─ Detection time (inverse)
└─ Coverage completeness

RESPONSE (0-1):
├─ Response time (inverse)
├─ Containment effectiveness
├─ Communication speed
└─ Coordination quality

RECOVERY (0-1):
├─ Recovery time objective (RTO) achievement
├─ Recovery point objective (RPO) achievement
├─ Data integrity preservation
└─ Service restoration speed
```

SECTION 9: SECURITY_USS SPECIFICATION NEW

9.1 Definition

Security_USS — 8% of CSI:

```
Security_USS = (Cyber + Physical + Operational + Resilience) / 4

CYBER SECURITY (0-1):
├─ Cyber_USS (see 8.3)
├─ AGI-level defense effectiveness
├─ Zero-trust implementation
└─ Encryption coverage

PHYSICAL SECURITY (0-1):
├─ Facility protection score
├─ Access control effectiveness
├─ Geographic distribution
└─ Hardening level

OPERATIONAL SECURITY (0-1):
├─ Process compliance
├─ Personnel vetting
├─ Information handling
└─ Incident management

RESILIENCE (0-1):
├─ Phoenix Protocol readiness
├─ Recovery capability
├─ Redundancy adequacy
└─ Continuity assurance
```

9.2 Thresholds

Security_USS	Level	Status	Action
0.90-1.00	□ L0	EXCELLENT	Maintain
0.80-0.89	□ L1	GOOD	Standard monitoring
0.65-0.79	□ L2	WARNING	Security review
0.50-0.64	□ L3	DANGER	Urgent hardening
<0.50	□ L4	CRITICAL	Emergency lockdown

Note: Security has HIGHER thresholds than other metrics — security-critical.

9.3 Current Global Estimate

```
Security_USS ≈ 0.62 (WARNING zone)

Components:
├─ Cyber: 0.68
├─ Physical: 0.70
├─ Operational: 0.55
└─ Resilience: 0.55

Primary gaps:
├─ Operational procedures immature
├─ Resilience not fully tested
└─ Target 2027: Security_USS ≥ 0.80
```

SECTION 10: TDI_USS SPECIFICATION □ NEW

10.1 Definition

TDI (Technical Debt Index) — Inverse Metric:

```
TDI_USS = 1 - TDI_raw

Where TDI_raw = accumulated technical debt (0-1 scale)

TDI_raw = (Code_Debt + Arch_Debt + Doc_Debt + Test_Debt) / 4

CODE DEBT (0-1):
├─ Outdated dependencies
├─ Code complexity
├─ Duplication ratio
└─ Style violations

ARCHITECTURE DEBT (0-1):
├─ Coupling level
├─ Modularity violations
├─ Scalability constraints
└─ Performance bottlenecks

DOCUMENTATION DEBT (0-1):
├─ Undocumented components
├─ Stale documentation
├─ Missing specifications
└─ Unclear interfaces

TEST DEBT (0-1):
├─ Test coverage gaps
├─ Flaky tests
```

├─ Missing integration tests
└─ Automation gaps
```

### 10.2 Interpretation

```
TDI_USS = 1 - TDI_raw

HIGH TDI_USS (>0.80) = LOW technical debt = GOOD
LOW TDI_USS (<0.50) = HIGH technical debt = BAD

Example:
├─ TDI_raw = 0.3 (30% debt)
├─ TDI_USS = 1 - 0.3 = 0.70 (WARNING)
└─ Action: Technical debt reduction plan
```

10.3 Thresholds

TDI_USS	Level	Status	Action
0.85-1.00	□ L0	EXCELLENT	Maintain discipline
0.70-0.84	□ L1	GOOD	Routine maintenance
0.55-0.69	□ L2	WARNING	Debt reduction sprint
0.40-0.54	□ L3	DANGER	Major refactoring
<0.40	□ L4	CRITICAL	Architecture review

10.4 Technical Debt Prevention

```
DESIGN PRINCIPLES:
├─ Pay debt immediately when small
```
```

├─ Never exceed TDI_raw > 0.30 (TDI_USS > 0.70)

├─ Quarterly debt reviews

├─ 20% time for debt reduction

└─ No new features if TDI_USS < 0.55

MONITORING:

├─ Automated TDI calculation (weekly)

├─ Trend tracking (monthly report)

├─ Alert on TDI_USS decline > 0.05/month

└─ Dashboard integration

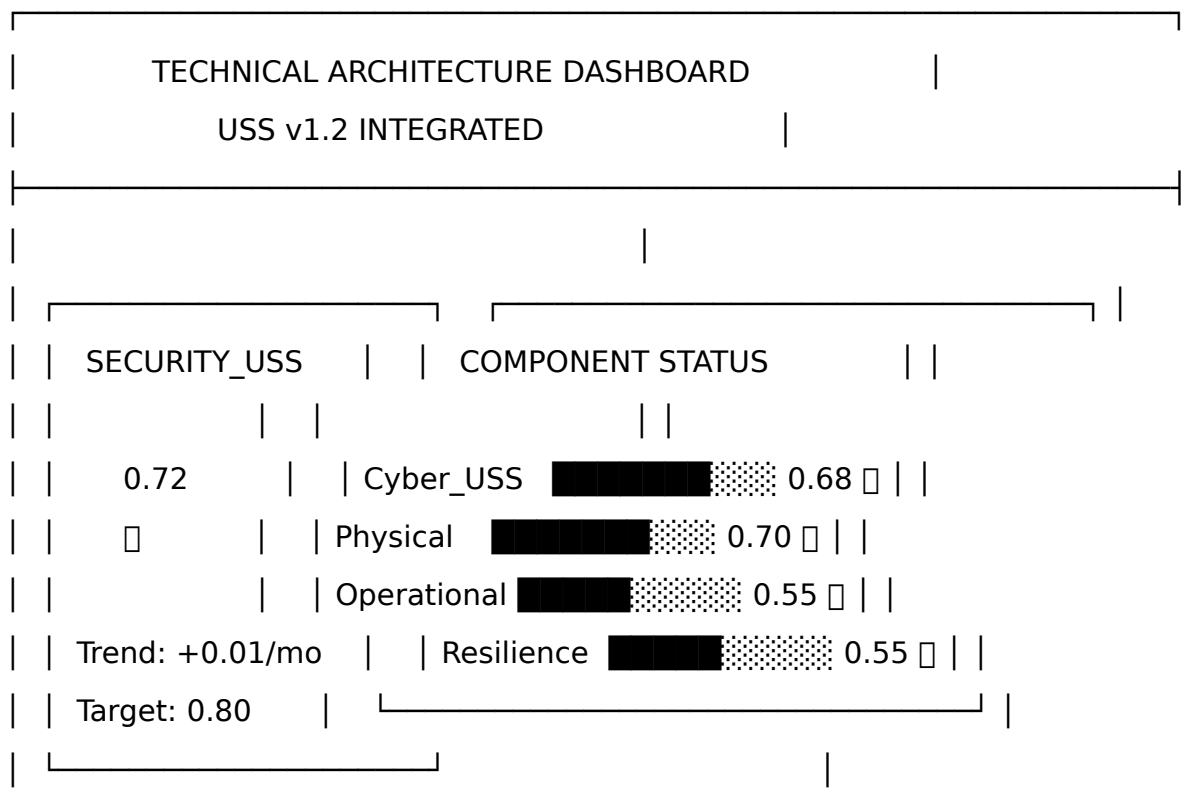
```

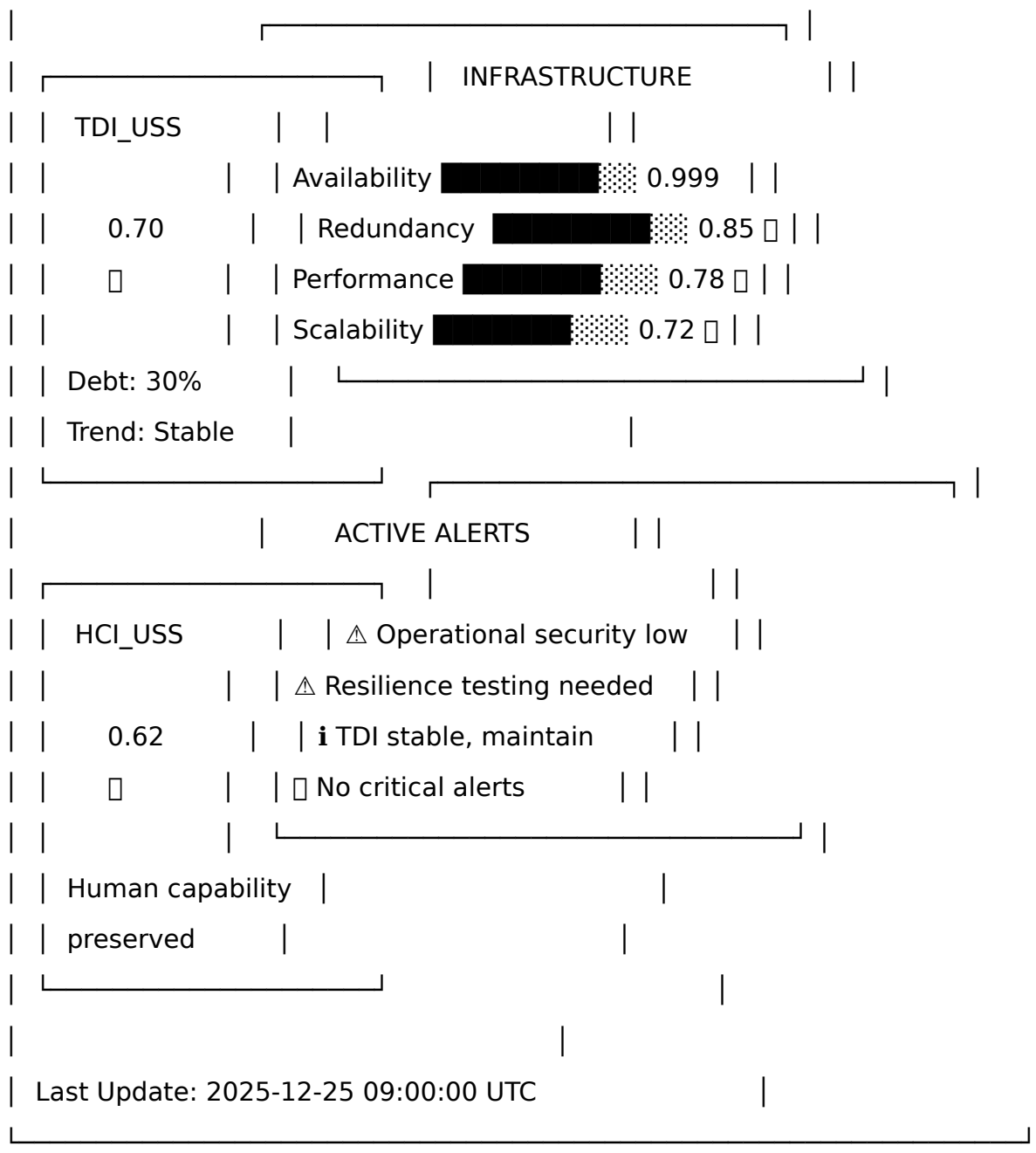
---

## SECTION 11: TECHNICAL DASHBOARD □ NEW

### 11.1 Dashboard Layout

```






```

### 11.2 Alert Configuration

```
TECHNICAL ALERTS (USS-Based):

SECURITY ALERTS:
├─ Security_USS < 0.65: WARNING
├─ Security_USS < 0.50: DANGER → Lockdown consideration
```
```

├─ Any component < 0.40: Component-specific alert
└─ Trend decline > 0.05/month: Investigation

TDI ALERTS:
├─ TDI_USS < 0.55: WARNING → Debt sprint required
├─ TDI_USS < 0.40: DANGER → Feature freeze
├─ TDI_USS decline > 0.10/quarter: Trend alert
└─ Any debt category > 0.50: Category alert

INFRASTRUCTURE ALERTS:
├─ Availability < 0.999: Immediate investigation
├─ Any component < 0.70: Component alert
├─ Performance degradation > 10%: Performance alert
└─ Capacity < 20% headroom: Scaling alert

HCI ALERTS (from architecture):
├─ HCI_USS < 0.50: WARNING → Reduce AGI assistance
├─ HCI_USS < 0.30: CRITICAL → Restructure symbiosis
└─ HCI decline > 0.05/year: Trend alert
```

---

## SECTION 12: HCI INTEGRATION IN ARCHITECTURE □ NEW

### 12.1 Human Capability Index (HCI_USS) Monitoring

**Architecture responsibility for HCI:**

```
TIERED ASSISTANCE (from Section 7) → HCI Impact:

TIER 1: FULL AUTONOMY

├─ AGI handles completely
├─ HCI impact: Neutral (no human involvement)
└─ Monitor: Ensure not expanding

TIER 2: COLLABORATION
├─ AGI + Human together
├─ HCI impact: Positive (skill development)
└─ Monitor: Balance maintained

TIER 3: HUMAN PRIMARY
├─ Human leads, AGI assists
├─ HCI impact: Highly positive
└─ Monitor: Sufficient challenges

TIER 4: HUMAN ONLY
├─ No AGI involvement
├─ HCI impact: Maximum development
└─ Monitor: Not shrinking
```

12.2 Architecture HCI Targets

```
ARCHITECTURE MUST ENSURE:

HCI_USS ≥ 0.60 (minimum)
├─ Problem-solving opportunities preserved
├─ Decision authority maintained
├─ Uncertainty tolerance exercised
├─ System thinking developed
└─ Challenge-seeking encouraged

TIER DISTRIBUTION TARGET:

├─ Tier 1 (Full Autonomy): ≤40% of interactions
├─ Tier 2 (Collaboration): 30-40%
├─ Tier 3 (Human Primary): 15-25%
├─ Tier 4 (Human Only): ≥5%
└─ Trend: Tiers 3-4 not decreasing
```

12.3 HCI Alert Thresholds

HCI_USS	Level	Status	Architecture Action
≥0.70	□	Healthy	Maintain tier balance
0.55-0.69	□	Warning	Increase Tier 3-4
0.40-0.54	□	Danger	Mandatory challenge programs
<0.40	□	Critical	Symbiosis restructuring

APPENDIX A: USS CONVERSION FOR TECHNICAL METRICS

Standard Conversions

```python
def convert_technical_to_uss(metrics: dict) -> dict:
    """Convert technical metrics to USS."""

    uss = {}

    # Direct conversions (already 0-1)
    for key in ['availability', 'redundancy', 'performance']:
        uss[key] = metrics.get(key, 0.5)

    # Inverse conversions (lower = better → higher USS)

```
    uss['tdi'] = 1 - metrics.get('tdi_raw', 0.3)
    uss['drift'] = 1 - metrics.get('drift_raw', 0.1)
    uss['latency'] = 1 - min(metrics.get('latency_ms', 100) / 1000, 1)

    # Percentage conversions
    uss['uptime'] = metrics.get('uptime_percent', 99.9) / 100
    uss['coverage'] = metrics.get('test_coverage', 80) / 100

    # Composite: Security_USS
    uss['security'] = (
        uss.get('cyber', 0.5) * 0.25 +
        uss.get('physical', 0.5) * 0.25 +
        uss.get('operational', 0.5) * 0.25 +
        uss.get('resilience', 0.5) * 0.25
    )

    return uss
```

Conversion Table

Raw Metric	Scale	USS Formula	Example
Uptime %	0-100%	v/100	99.9% → 0.999
TDI	0-1 (debt)	1-v	0.3 → 0.70
Latency ms	0-1000	1-(v/1000)	100ms → 0.90
Coverage %	0-100	v/100	80% → 0.80
Drift	0-1	1-v	0.1 → 0.90
Risk score	0-10	1-(v/10)	3 → 0.70

REFERENCES

- ANNEX Metrics Framework v3.3 — USS/CSI specification
- V12 Measurement Systems v1.1 — CSI formula
- Security Architecture v3.0 — Technical details
- Integration Map v1.6 — System architecture

END V04 v1.2 — USS/CSI INTEGRATED

Critical Reminders:
- Security_USS is 8% of CSI — critical component
- TDI_USS monitors technical debt (inverse metric)
- HCI_USS must be preserved by architecture design
- All technical metrics in USS for unified monitoring
- Dashboard enables automated alerting